AFRL-IF-RS-TR-2005-132
Final Technical Report
April 2005

INFRASTRUCTURE OPERATIONS TOOLS ACCESS (IOTA)/TRUSTED TRANSFER AGENT (TTA)

Northrop Grumman

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-132 has been reviewed and is approved for publication

APPROVED:

RICHARD J. LORETO
Project Engineer

FOR THE DIRECTOR:

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	APRIL 2005	Final Nov 03 – Sep 04

4. TITLE AND SUBTITLE
INFRASTRUCTURE OPERATIONS TOOLS ACCESS (IOTA)/TRUSTED TRANSFER AGENT (TTA)

5. FUNDING NUMBERS
C - F30602-03-D-0026/0004
PE - 63260F
PR - 3481
TA - QP
WU - 04

6. AUTHOR(S)
Jim Muller

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northrop Grumman
7902 Turin Road
Rome New York 13440

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFEB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-132

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard J. Loreto/IFEB/(315) 330-3793/ Richard.Loreto@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*
Northrop Grumman provides a description of the technical architecture of IOTA, with a focus on the security architecture (TTA) and the initial instantiation of IOTA in the ISAIAH prototype. The report describes IOTA web services for information handling, transformation, and management and the component framework helps support the Air Force implementation of the Net-Centric Enterprises Services (NCES) and the larger Global Information Grid Enterprise Services (GIGES). The method utilized in the report is technically descriptive presenting views of the software architecture with accompanying narrative. The presentation of the ISAIAH prototype provides an example of the application of the IOTA infrastructure within a mission application environment.

14. SUBJECT TERMS
Net-Centric Enterprises Services, Web Services, Information Services, Publish-Subscribe, Data Access Infrastructure, Information Discovery, Information Assurance/Security, Intelligence Data Handling, Cross-Security Domain Services

15. NUMBER OF PAGES
36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION AND PURPOSE

This report, entitled *Infrastructure Operations Tools Access -Trusted Transfer Agent (IOTA-TTA) Final Technical Report,* documents the software architecture, summary and recommendations resulting from the IOTA-TTA effort. This report is being provided under Contract No. F30602-03-D-0026/0004 and satisfies CDRL A005.

1.1 Document Roadmap

Section 2 of this report describes the IOTA v1.0 system developed under this task. Section 3 describes the Isaiah prototype fielded at the 480IW. Section 4 itemizes the documentation produced under this task. Section 5 lists appropriate references. Section 6 provides a glossary of terms.

2. IOTA OVERVIEW

2.1 IOTA Mission

IOTA provides services for information handling, transformation, and management for use by a variety of mission applications. The IOTA architecture is based on a component framework that allows services and service capabilities to be easily added or enhanced. IOTA follows guidelines for the Global Information Grid Enterprise Services (GIG ES).

2.2 IOTA Purpose and Scope

IOTA provides access to information from a variety of data sources through Web services and publish-subscribe mechanisms. Information provided by IOTA falls into three categories: imagery, messages and reports, and general military Intelligence. IOTA services and publish-subscribe mechanisms can be used by applications to obtain information with standard tagged metadata from these sources. Currently applications planning to use IOTA services are the Joint Targeting Toolbox, JTT, Automated Assistance with Intelligence Preparation of the Battlespace (A2IPB) and the Isaiah Web application deployed at 480IW.

2.3 System Description

This section describes the IOTA software interfaces to other software applications. It describes both the interfaces required by applications to obtain information products from IOTA and the interfaces IOTA uses to obtain data from other applications.

2.3.1 IOTA and GIG ES

IOTA represents the Air Force implementation of portions of the Net-Centric Enterprises Services (NCES). NCES is the proposed program for the development and operation of the 9 Core Enterprise Services (CES) of the larger Global Information Grid Enterprise Services (GIG ES) as depicted in Figure 1-1.

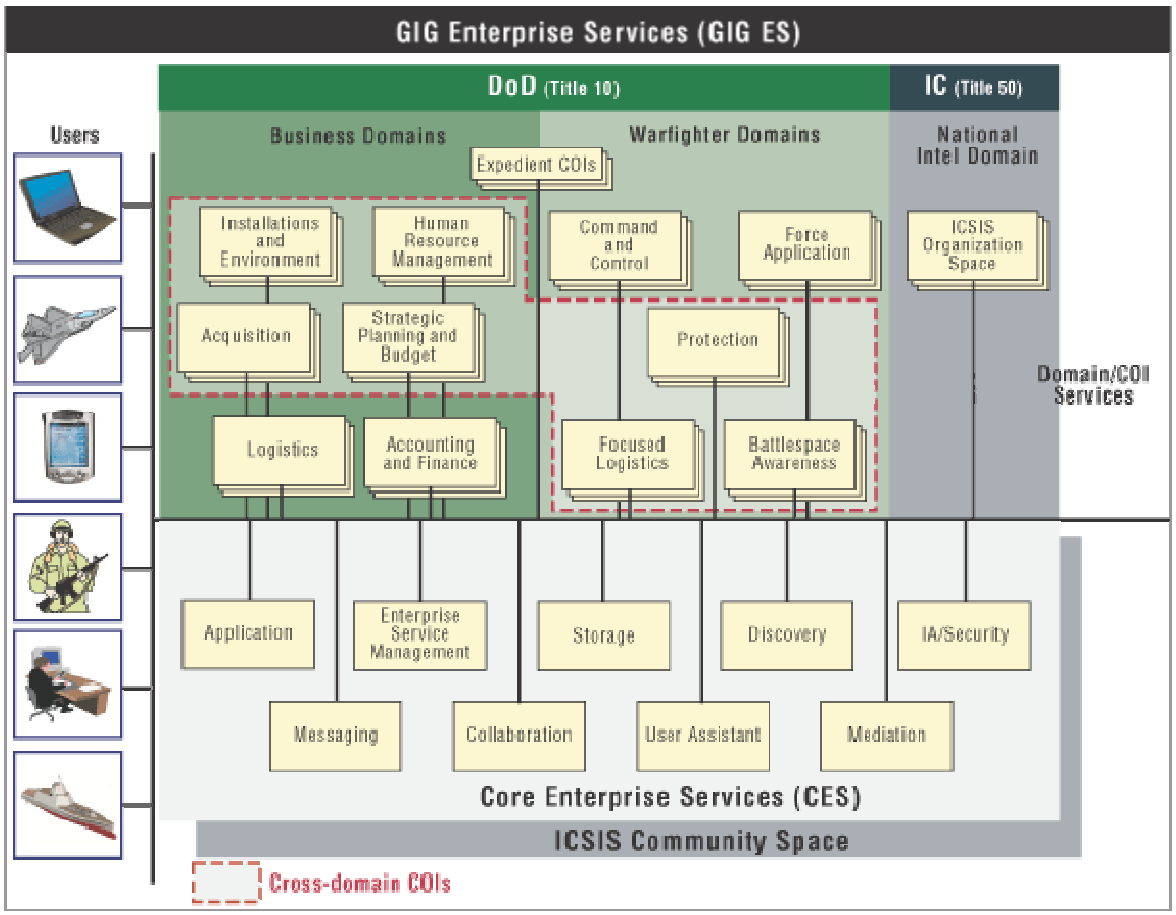

Figure 1-1: GIG ES Architecture

The CES consists of services for:

- *Enterprise Service Management* - The set of services that enable the life cycle management of the information environment and supports the performance of the NetOps activities necessary to operationally manage information flows in the information environment.
- *User Assistant* - Automated capabilities that learn and apply user preferences and patterns to assist users to efficiently and effectively utilize GIG resources in the performance of tasks.
- *Messaging* - Provides services to support synchronous and asynchronous information exchange.
- *Information Assurance/Security* - The set of services that provide a layer of defense in depth to enable the protection, defense, integrity, and continuity of the information environment and the information it stores, processes, maintains, uses, shares, disseminates, disposes, displays, or transmits.
- *Discovery* - The set of services that enable the formulation and execution of search activities to locate data assets (e.g., files, databases, services, directories, web pages, streams) by exploiting metadata descriptions stored in and or generated by IT repositories (e.g., directories, registries, catalogs, repositories, other shared storage).

- *Storage Services* - The set of services necessary to provide on demand posting, storage and retrieval of data
- *Mediation* - The set of services that enable transformation processing (translation, aggregation, integration), situational awareness support (correlation and fusion), negotiation (brokering, trading, and auctioning services) and publishing
- *Collaboration* - The set of services that allows users to work together and jointly use selected capabilities on the network (i.e., chat, online meetings, work group software etc.)
- *Applications* - The set of services necessary to provision, host, operate and manage the GIG ES assured computing environment

IOTA addresses services for Information Discovery, Messaging, Enterprise Service Management, Mediation, and Information Assurance/Security, in addition to providing services for information visualization and information product production support. IOTA information services provide robust and secure access to information products needed by mission applications.

IOTA is designed to meet the objectives of the GIG ES and NCES. That is:

- Deliver capabilities-based service infrastructure for ubiquitous access to timely, secure, decision quality information by edge users
- Enable information providers to post any information they hold
- Enable edge users to rapidly and precisely discover and pull information resources
- Provide security for, and coordinated management of, netted information resources
- Provide data interoperability versus application interoperability

IOTA provides standards-based Web services for obtaining integrated information products from information sources and repositories and provides robust publish and subscribe capabilities. IOTA provides capabilities for cross-security domain services through the Information Support Server Environment (ISSE) STAR Guard capability and leverages secure platform services through the Joint Enterprise DoDIIS Infrastructure (JEDI). Additionally, IOTA provides the community with:

- Interoperability through IC standards for interfaces between components and platforms
- Architecture/Infrastructure – a standards-based framework with common Application Program Interfaces (APIs) for interchangeable Commercial-Off-The-Shelf (COTS)/Government-Off-The-Shelf (GOTS) components, services and applications
- Multi-INT tools for visualization, and temporal and geo-spatial analysis, including near-real time data, imagery, and reports
- Information dissemination through Web services, portal interfaces, and application-to-application interfaces
- System Administration through integration with standard platform services and tools for network and platform management
- Security tools and services for meeting accreditation and enterprise defense requirements

2.3.2 IOTA Version 1.0

Version 1.0 of IOTA provides Web services and publish and subscribe mechanisms to obtain imagery, messages, or general military intelligence information products from a variety of sources. The data sources available through IOTA services in version 1.0 are:

- Image Product Library (IPL 3.0.2 and 2.5.1)
- Demand Driven Direct Digital Dissemination (5D)
- GCCS I3 Imagery Service (ITS) 4.1.7.0
- Multimedia Message Manager (M3) 3.0
- Imagery Exploitation Support System (IESS) 4.2.x
- Information Extraction Tool (IET) 1.1
- Modernized Integrated Database (MIDB 2.0 and 2.1)
- Configuration Support Processor (CSP 5.9.1)

IOTA services are components that can be deployed as needed. Only the services required for a particular IOTA installation are deployed with it. Additional services may be deployed later if needed. Figure 1-2 provides a high level view of the IOTA version 1.0 architecture.

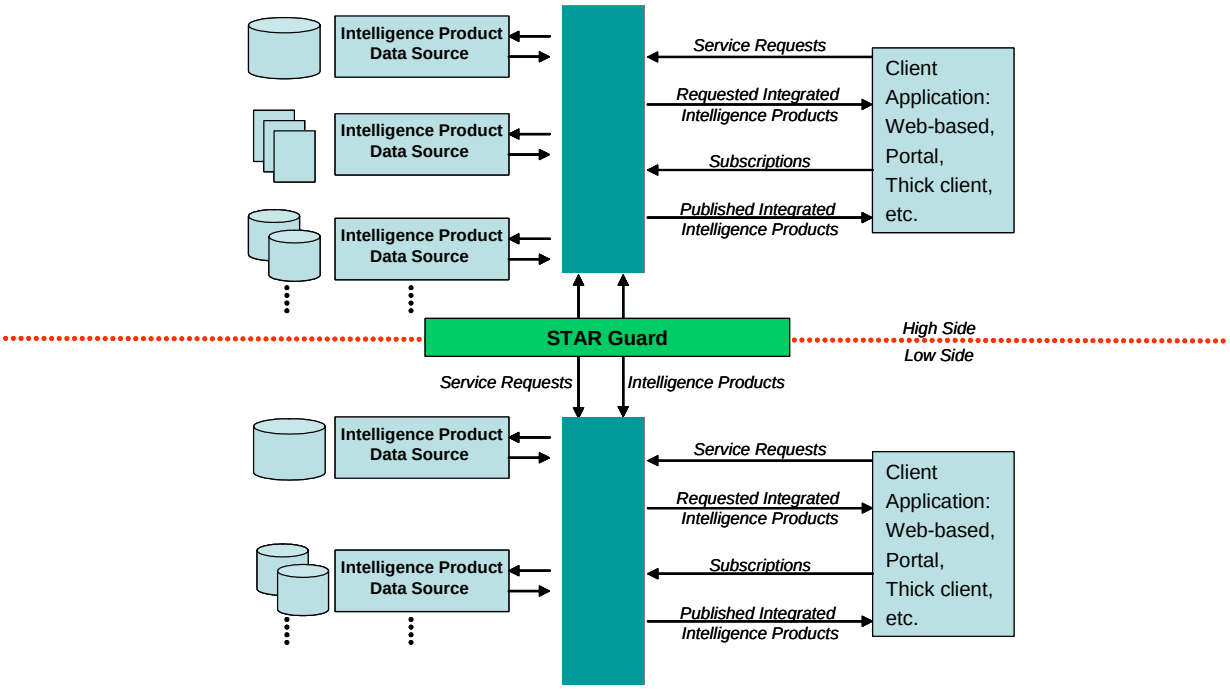

Figure 1-2: IOTA High-level Architecture

IOTA services are designed to support multiple and varying kinds of client interface requirements, and cross-boundary access via the ISSE STAR Guard capability. As Figure 1-2 shows, client applications can use IOTA to invoke services or to set up subscriptions. IOTA services provide integrated information products from a variety of data sources. The same information products can be obtained through the client application establishing a subscription. In that case, any new information product that becomes available through a single or combined set of information from data sources is automatically provided to the client application.

Figure 1-3 provides a more detailed view information flow among the key IOTA architectural components.

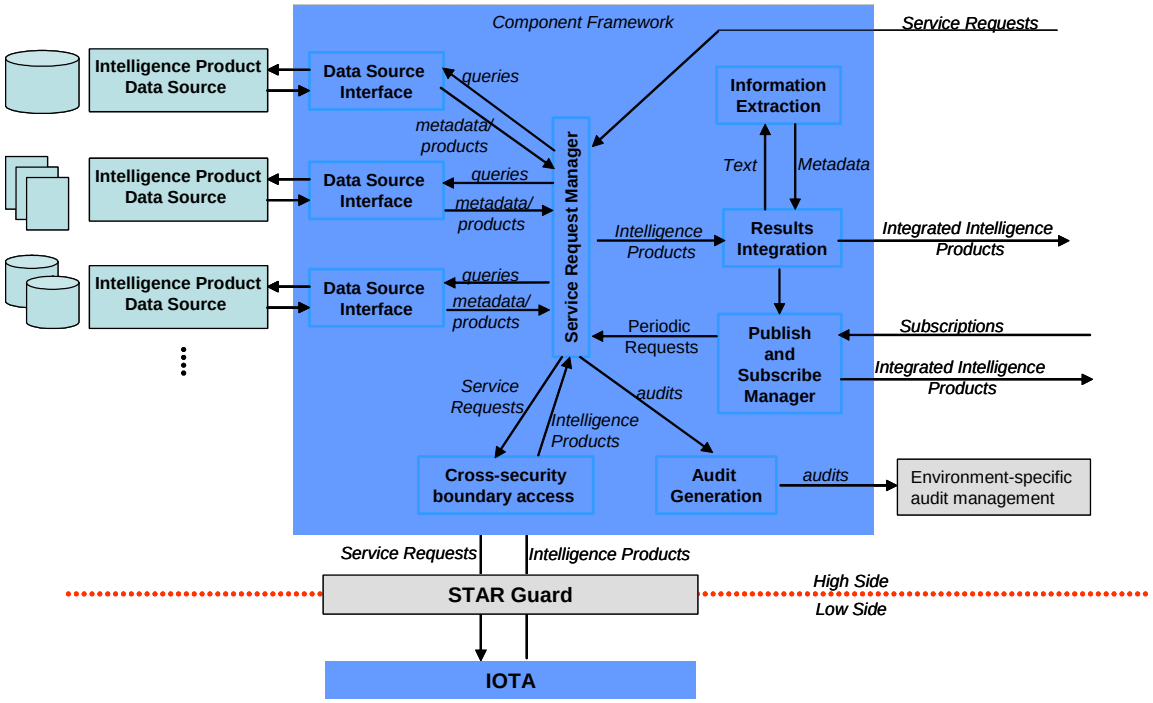

Figure 1.3: IOTA Architecture Information Flow

Service requests are handled by the Service Request Manager and result in one or more invocations of Data Source Access Component interfaces. Results are combined into integrated products by the Results Integration component using a standard set of metadata. Subscription requests are handled by the Publish and Subscribe Manager. Publisher processes make periodic requests from data sources to check for new information products. When new information products are discovered, the publish and subscribe components exercise a call back function provided by the subscribing client to deliver the information products.

The Information Extraction component is used when services require that metadata be extracted from text data. For example, report or message text containing a BE number or Mission Id would be processed through the data extraction component if the data source did not already provide this information as metadata.

Audits are generated by the service request manager, and fed to the underlying audit data handling mechanism. For example, on Sun Solaris and other UNIX platforms, IOTA would be configured to write audits using **`syslog`** functions. On Windows platforms the event log mechanism is used.

Cross-boundary services are managed by an interface to the ISSE Guard STAR Guard capability. Service requests on one side of the boundary are passed to the IOTA residing on the other side for processing and the results returned through the STAR Guard.

The services implemented by IOTA version 1.0 are:

- Information Dissemination

- Product Requests
  - Integrated product metadata search ("find all reports and imagery about this range of BE numbers and return metadata - date, location, source, other collection missions, etc.")
  - Integrated product request ("return all integrated products containing reports and imagery matching some criteria")
  - Single type product metadata request (e.g., - find all imagery metadata meeting some metadata criteria)
  - Single type product request (e.g., - return all imagery matching some criteria)
  - Single source product metadata search (e.g., - find all imagery metadata from IPL 3.0 meeting some metadata criteria)
  - Single source product request (e.g., - return all imagery from IPL 3.0 meeting some metadata criteria)
- Product Subscription
  - Subscribe to integrated product metadata
  - Subscribe to integrated products
  - Subscribe to single type product metadata
  - Subscribe to single type product
  - Subscribe to single source product metadata
  - Subscribe to single source products

- Information Discovery
  - List sources available ("list the known data sources")
  - List product types available ("list the known available product types: e.g., result would be NITF images, MPEG2 videos, IPIRs, INTSUMs")
  - Subscribe to product types available ("tell me when a new type is added")
  - Subscribe to sources available ("periodically report all active data sources" or "when a new data source is added, push its description to me")

2.3.3 IOTA Development Process

IOTA development is organized into three spirals. Spiral 1 will provide single source information product services and a publish and subscribe mechanism for IPL, 5D, GCCS I3 ITS, M3, IET, MIDB and IESS. Spiral 2 will provide integrated product services and cross-boundary services. Spiral 3 will focus on tools and integrated information management capabilities and will also address Universal Description Discovery and Integration (UDDI) registration of IOTA services. This task covers the activities of Spiral one culminating in BETA I testing within the Joint Integrated Test Facility (JITF).

2.3.4 System Functional Description

IOTA uses a service-oriented architecture for secure access to data sources. The IOTA application components run within an application server. IOTA components can run under two application services: either JBoss version 3.2.3 or WebLogic version 8.1. Both of these application servers run on either Windows or Solaris platforms. Table 3.1 shows the relationship of the functional components of the IOTA architecture.

Functional Component	Purpose
Service Request Manager	The Service Request Manager handles all external requests for services other than publish/subscribe requests. It generates audits via the Audit Generation service, as described later in this document. It routes requests to various configured data sources (via the appropriate Data Source Interface)
Publish & Subscribe Manager	This component handles external publish/subscribe requests, and audits startup, shutdown, request for subscription, and end of subscription events via the Audit Generation service. It subscribes to or periodically queries the appropriate data source(s) for new data.
Audit Generation	This component allows IOTA to abstract itself from the system's auditing requirements, allowing integration with either the Windows Event Viewer or the Solaris *syslog* capability. It takes audit information from the Service Request Manager and the Publish & Subscribe Manager and converts them to the system-specific formats needed.
Results Integration	This component gathers together the results from requests made by the Service Request Manager, packages them together, and delivers them to the user's configured location(s).
Information Extraction	This component takes raw text messages found in a data source and parses out applicable metadata, such as BE Numbers, dates, and the like.
Data Source Interface	Data source Interface components are implemented for each data source IOTA interfaces to. These components negotiate the data source API to obtain information needed to answer ITOA service requests.

Table 3.1: Functional Components

The services implemented by IOTA version 1.0 are:

- Information Dissemination
  - Product Requests
    - Integrated product metadata search ("find all reports and imagery about this range of BE numbers and return metadata - date, location, source, other collection missions, etc.")
    - Integrated product request ("return all integrated products containing reports and imagery matching some criteria")
    - Single type product metadata request (e.g., - find all imagery metadata meeting some metadata criteria)
    - Single type product request (e.g., - return all imagery matching some criteria)
    - Single source product metadata search (e.g., - find all imagery metadata from IPL 3.0.4 meeting some metadata criteria)
    - Single source product request (e.g., - return all imagery from IPL 3.0.4 meeting some metadata criteria)
  - Product Subscription
    - Subscribe to integrated product metadata
    - Subscribe to integrated products

        - Subscribe to single type product metadata
        - Subscribe to single type product
        - Subscribe to single source product metadata
        - Subscribe to single source products
- Information Discovery
    - List sources available ("list the known data sources")
    - List product types available ("list the known available product types: e.g., result would be NITF images, MPEG2 videos, IPIRs, INTSUMs")
    - Subscribe to product types available ("tell me when a new type of product is added")
    - Subscribe to sources available ("periodically report all active data sources" or "when a new data source is added, push its description to me")

2.3.5 IOTA Application Interfaces

This section describes the IOTA "client" interfaces. In this context, a client is any application that is invoking one or more IOTA services. These interfaces specify how applications use IOTA to obtain information products. IOTA provides two kinds of interfaces for obtaining information products: a request interface and a subscription interface. The same information products are available through both the request interface and the subscription interface; the difference is that the request interface is used to return products available at the time services are invoked, whereas the subscription interface allows applications to continuously obtain information products as they become available.

The request interface is a Web service based interface, described via an IOTA Web service definition language (WSDL) document. This interface uses a request/response paradigm; each service request results in exactly one response message. This response message may contain product metadata, links to external resources, status information or links that invoke other services to get supplementary information.

The subscription interface allows the specification of a "standing" request. This interface uses a subscribe/listen paradigm; a client subscribes for new information products based upon a set of criteria, and whenever the IOTA server becomes aware of "new" information this information is asynchronously provided to the subscriber via a method callback. The information payload passed to the callback method contains the same information as that passed in the Web service invocation response. The current implementation of this capability is built upon a Java Messaging Service (JMS) backbone.

The request interface allows retrospective queries for information products from the time that the query is made. The subscription interface allows for obtaining information products from the time a subscription is established until the time it is ended. In either interface mode, applications must provide user credentials to IOTA for authentication. IOTA authentication mechanisms are described in Section 3.1.

The same parameters are used by applications to invoke services and establish subscriptions. The same information objects are delivered in response to a service request, or published for subscribers. The parameters used as input for requests or subscriptions are objects defined by the IOTA input types. The IOTA output types are the objects for returned information. IOTA input types are described in Section 3.2. IOTA output types are described in Section 3.3.

Applications need only use the services they require, so only the IOTA services required will be deployed at a given site. IOTA is designed so that as requirements change, service components can easily be added to the IOTA framework to provide additional capability.

2.3.6 IOTA Configurations

The tables below detail the three IOTA configurations developed under this task and accredited through JITF BETA I Testing.

Configuration Table – IOTA PC Server utilizing JBoss Application Server

Configuration Item	Server
Platform Type (Minimum)	2 GHz CPU
Platform Type (Recommended)	Dual 2 GHz CPU
Peripheral Devices	CDROM
OS version	Windows 2000 Professional
OS modules/packages	See System Installation Guide
Required OS Patches	See System Installation Guide
Minimum Memory	2 GB
Recommended Memory	2 GB +
Disk Space for IOTA Binaries	1 G
Disk Space for IOTA Work Space	10 GB
Disk Space for MySQL Database	1 G
Network Configuration	10/100 Ethernet
Partitions	See System Installation Guide
Software Packages	Java™ 2 SDK, Standard Edition Version 1.4.1/1.4.2: Apache Web Server 1.3.19 Apache eXtensible Interaction System, AXIS 1.1, Java API for XML-based RPC JAX-RPC 1.0 SOAP with Attachments API for Java, SAAJ 1.1 Xerces XML Parser 2.5.0 Castor binding framework, Castor 0.9.5.2 XQEngine XQuery/XPath parser 0.61 Joint Battlespace Infosphere, JBI 1.1 MySQL 4.0.18 JBoss Application Server 3.2.5 Bundled with Tomcat servlet container 5.0.26
Additional Information	None
Configured Sources (Plugins)	M3 3.1, IESS 4.2*, IPL 2.5.1 PATCH 4, IPL 3.0.4, IPL 3.5, MIDB 2.0 (Yorkshire), MIDB 2.1 (Everest-DoDIIS), CSP 5.9.1

Configuration Table – IOTA Solaris server utilizing WebLogic Application Server

Configuration Item	Server
Platform Type (Minimum)	Ultra 60/Dual USparc IIi 400 MHz CPU
Platform Type (Recommended)	SunBlade Dual USparc III CPU
Peripheral Devices	CDROM
OS version	Solaris 8 64 bit
OS modules/packages	See System Installation Guide
Required OS Patches	See System Installation Guide
Minimum Memory	2 GB
Recommended Memory	2 GB +
Disk Space for IOTA Binaries	1 G
Disk Space for IOTA Work Space	10 GB
Disk Space for Sybase Database	1 G
Network Configuration	10/100 Ethernet
Partitions	See System Installation Guide
Software Packages	Java™ 2 SDK, Standard Edition Version 1.4.1/1.4.2: Apache Web Server 1.3.19 Apache eXtensible Interaction System, AXIS 1.1, Java API for XML-based RPC JAX-RPC 1.0 SOAP with Attachments API for Java, SAAJ 1.1 Xerces XML Parser 2.5.0 Castor binding framework, Castor 0.9.5.2 XQEngine XQuery/XPath parser 0.61 Joint Battlespace Infosphere, JBI 1.1 Sybase 12.5.1 WebLogic 8.1
Additional Information	None
Configured Sources (Plugins)	M3 3.1, IESS 4.2*, IPL 2.5.1 PATCH 4, IPL 3.0.4, IPL 3.5, MIDB 2.0 (Yorkshire), MIDB 2.1 (Everest-DoDIIS), CSP 5.9.1

Configuration Table – IOTA Solaris server utilizing JBoss Application Server

Configuration Item	Server
Platform Type (Minimum)	Ultra 60/Dual USparc IIi 400 MHz CPU
Platform Type (Recommended)	SunBlade Dual USparc III CPU
Peripheral Devices	CDROM
OS version	Solaris 8 64 bit
OS modules/packages	See System Installation Guide
Required OS Patches	See System Installation Guide
Minimum Memory	2 GB
Recommended Memory	2 GB +
Disk Space for IOTA Binaries	1 G
Disk Space for IOTA Work Space	Additional storage space up to10 GB and same location
Disk Space for Sybase Database	1 G
Network Configuration	10/100 Ethernet
Partitions	See System Installation Guide
Software Packages	Java™ 2 SDK, Standard Edition Version 1.4.1/1.4.2: Apache Web Server 1.3.19 Apache eXtensible Interaction System, AXIS 1.1, Java API for XML-based RPC JAX-RPC 1.0 SOAP with Attachments API for Java, SAAJ 1.1 Xerces XML Parser 2.5.0 Castor binding framework, Castor 0.9.5.2 XQEngine XQuery/XPath parser 0.61 Joint Battlespace Infosphere, JBI 1.1 JBoss Application Server 3.2.5 Bundled with Tomcat servlet container 5.0.26 Sybase 12.5.1
Additional Information	None
Configured Sources (Plugins)	None

2.4 IOTA-TIGER Software Architecture

Figure 2-1 depicts the high-level software architecture that was designed under this effort for the IOTA-TIGER system. Its purpose is to enable users in the low security domain access to information in the high-side security domain via the IOTA capabilities and its established methods for publish and subscribe-based information dissemination. It should be noted that this architecture description specifically does not address the reverse situation in which users in the high security domain are provided access to information in the low-side security domain. It was, however, an implementation objective to design the initial version of the software in such a way as to handle this additional user need through configuration changes only (i.e., little or no software change required).

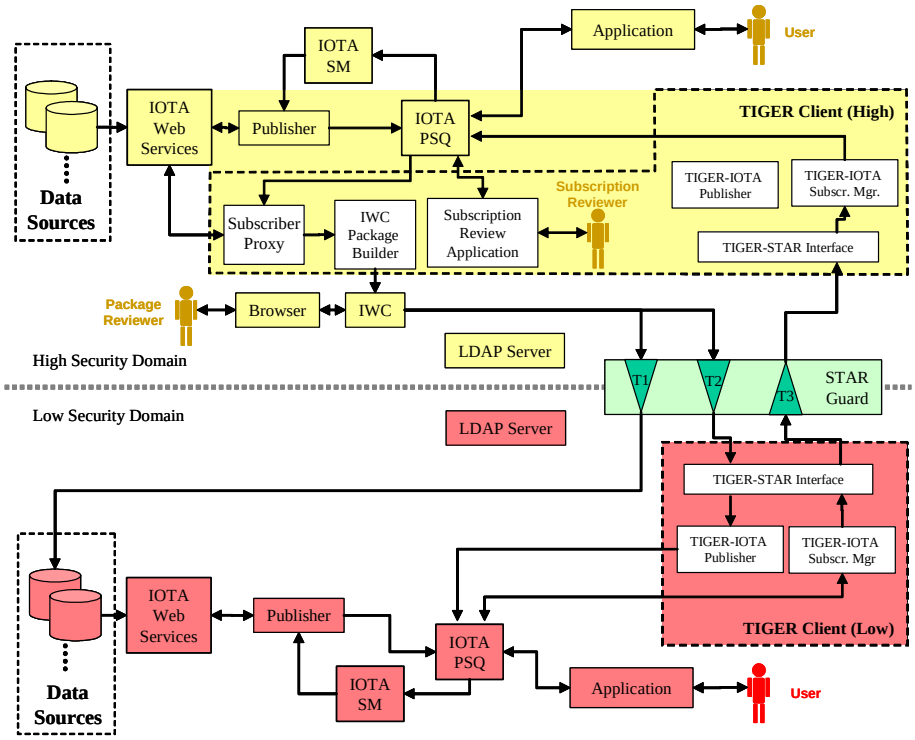

Figure 2-1: IOTA-TIGER Software Architecture

The applications, data sources, IOTA Web Services, IOTA Publisher, IOTA Subscription Manager (SM) and IOTA Publish Subscribe Query (PSQ) components, depicted in Figure 2-1, are existing components of the IOTA architecture and thus are described in other IOTA design documentation and will not be described within this document.

Major software components depicted in the diagram that are new to the IOTA-TIGER architecture that will be described further include:

- ISSE Web Client (IWC): The IWC is an existing component of the ISSE Guard suite of applications. Its primary purpose is to ensure that a secure and completely reliable human review is executed by an authorized reviewer on information products queued for release across the ISSE Guard boundary control application; in this case, the STAR Guard. The user interface to this application is a series of web pages accessed via a standard COTS browser using mutually authenticated Secure Socket Layer (SSL).
- Browser: A standard COTS web browser is used to access the IWC-provided web pages used for the review and release of IOTA-published information and to access the Subscription Review Application used for the review, approval and instantiation of low-side subscription requests for high-side information.
- Secure Trusted Automated Routing (STAR) Guard: The STAR Guard is an existing component of the ISSE Guard suite of applications. Its primary purpose is to provide for the secure and high-speed transfer of information automatically across a security boundary. The STAR Guard architecture allows for the creation of multiple, unidirectional threads (either high-to-low or low-to-high) for processing specific types of information. For the IOTA-TIGER STAR Guard configuration, three threads are envisioned: T1 - a high-to-low thread to process published IOTA objects, including

product and metadata files(e.g., imagery products); T2 - a high-to-low thread to process published IOTA information object without attached files (e.g., formatted intelligence messages); and T3 - a low-to-high thread to process subscriptions and subscription cancellations.

- TIGER Client: The TIGER Client is a new software component that facilitates the flow of information (e.g., published information, subscription requests, subscription cancellation requests) between the IOTA Publish/Subscribe components in the high and low domains and the appropriate STAR Guard threads. Additional components which are part of the high-side TIGER Client only include:
  - Subscription Review Application (SRA): The SRA provides a queuing mechanism for managing low-side subscription requests on the high-side and allows a human reviewer on the high-side to examine and approve or reject each subscription before it is published into the high-side IOTA architecture.
  - Subscription Proxy: The Subscription Proxy is a new software component that provides an interface between the IOTA PSQ and the IWC Package Builder application. Its primary purpose is to act as a proxy for all low-side subscriptions, receiving published information in response to those subscriptions and initiating the process for transferring them across the security boundary.
  - IWC Package Builder: The IWC Package Builder queues IOTA-published information to the IWC for review and release. The IWC Package Builder receives IOTA information (to include IOTA provided information, metadata and supplementary files) from the subscription proxy, and creates and posts to IWC packages requiring review and release.

2.4.1 IOTA-TIGER Data Flows

In order to facilitate the ability for a low-side user to “reach up” into high-side resources via a publish and subscribe paradigm, three separate, but related, data flows were specified: low-to-high Subscription Data Flow, high-to-low Published Information Data Flow, and low-to-high Subscription Cancellation Data Flow.

Figure 2-2 depicts the enumerated steps comprising the low-to-high Subscription Data Flow. This flow allows a subscription generated by a low-side user/application to be propagated and instantiated via TIGER components into the high-side IOTA publish and subscribe components.

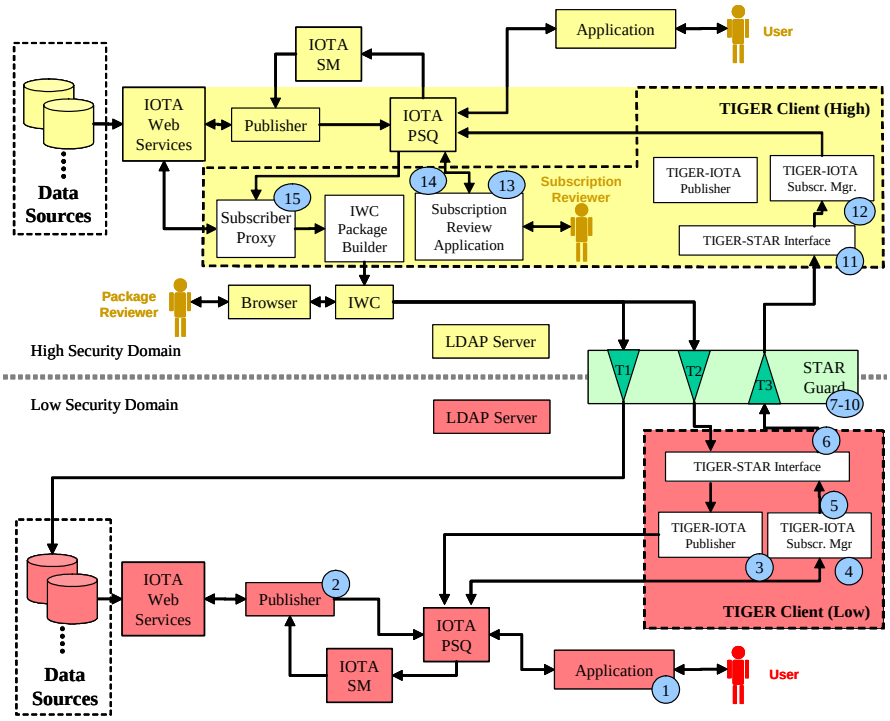

Figure 2-2: IOTA-TIGER Low-to-High Subscription Data Flow

The following describes, in detail, the processing occurring within each enumerated step shown in Figure 2-2:

1. An application in the low-side domain establishes a subscription. This involves opening a connection with the IOTA PSQ component, publishing an IOTA subscription information object, and establishing the subscription. (These steps are performed within the IOTA PSQ Application Programmer's Interface (API)). Each subscription established by a low-side application will need to be marked as a request for a high-side information/product and will need the credentials of the low-side users that are issuing the request added to the metadata of the information object.
2. If needed, the IOTA Subscription Manager on the low-side, establishes a low-side publisher. The application will receive any new information from low-side data sources. (This is normal IOTA operation).
3. The TIGER Application, on the low-side of the STAR Guard, will consist of a Subscription Manager. The Subscription Manager will register a callback with the IOTA PSQ for all IOTA subscription information objects.
4. The TIGER Subscription Manager will receive all subscription information objects as they are established and will evaluate the metadata in each subscription to verify which ones are requests for high-side information/products. Only those requesting high-side information/products will be allowed to continue on through the process.
5. All verified high-side information objects will be passed on to the STAR Guard Interface component of the TIGER Application. Each information object will consist of an eXtensible Markup Language (XML) message, which will contain the object's metadata with a payload.

6. The STAR Guard interface will then pass the information object on to the appropriate STAR Guard thread for evaluation and dissemination across the security boundary.
7. Each subscription object received by a STAR Guard thread will be scanned for viruses prior to any other activity. Failure will terminate the transfer of the object across the security boundary and the information object will be placed in quarantine.
8. If the information object passes the virus scan, it will then be parsed and verified against a previously loaded XML Schema. If the XML message fails to verify according to the schema, then the information object will be dropped and transfer will be terminated.
9. Once the parsing and verification process has been successfully completed, the credentials of the user who has initiated the subscription, will be verified for authenticity. Failure in this verification process will terminate the transfer of the information object across the security boundary.
10. The STAR Guard thread will apply all appropriate filters once the subscription is determined to contain a valid request.
11. After the filters have been applied, the XML information object will be re-packaged and forwarded on to the high-side TIGER Subscription Manager. Depending on security policy, it may be necessary to re-package the message using the STAR Guard credentials.
12. The high-side TIGER Subscription Manager receives the information object from the threads and publishes the subscription information object to the high-side IOTA. The IOTA PSQ metadata for the published information object indicates it is a low-side subscription information object.
13. A high-side application, the Subscription Review Application (SRA), has already been established to subscribe to low-side subscriptions published by TIGER.
14. The SRA queues the subscription for review. Once reviewed and approved, the SRA will place the subscription with the PSQ.
15. The client Subscription Proxy is created to receive information relative to the approved subscriptions and will receive notifications from the PSQ. It will then deposit the data for the IOTA-IWC Info Gateway (I3G) when it becomes available at some future time.

Figure 2-3 depicts the enumerated steps comprising the high-to-low Published Information Data Flow. This data flow allows information published by high-side data sources to be compared against established subscriptions (including those originating from a low-side user/application via the previous data flow). If a low-side subscription is satisfied, the information is then transferred to the low-side domain via a series of steps involving a reliable human review. Upon delivery to the low-side Data Source/IOTA PSQ, the low-side IOTA components notify the user/application of the existence of new information meeting their subscription.

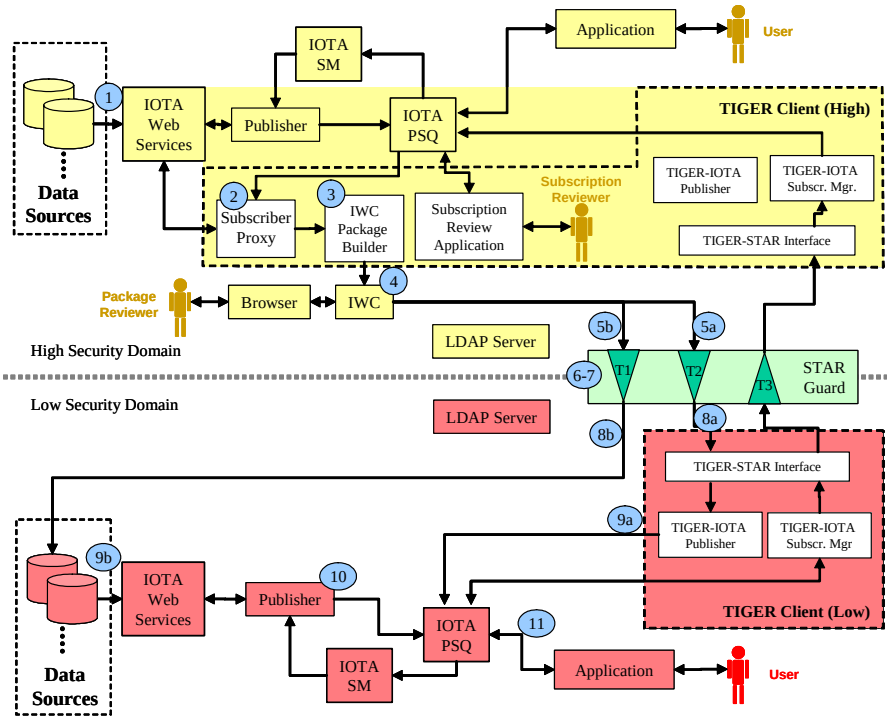

Figure 2-3: IOTA-TIGER High-to-Low Published Information Data Flow

The following describes, in detail, the process occurring within each enumerated step shown in Figure 2-3:

1. Information is published to a high-side data source. In turn, the PSQ notifies the Subscription Proxy of the new data.
2. The Subscription Proxy receives information from the PSQ and deposits the data for the IWC Package Builder (IPB) to receive. If those notifications reference products, the Subscription Proxy will order the products, along with their metadata, and deposit the data for the IPB to receive.
3. The IPB application receives either the ordered product and its metadata or the information object. The application then compiles them into a package and queues them for review.
4. The package reviewer then reviews and approves the package for release using a browser interface to the IWC.
5. Once reviewed and approved using the IWC, the high-side information/product packages are signed by the reviewer and sent to the appropriate high-side STAR Guard thread, which is either:
    a. A thread to handle formatted message data, or
    b. A thread to handle product files and their associated metadata
6. Once the thread receives the package from the interface, it will verify that the proper reviewer credentials are attached and that this package can pass across the security boundary in accordance with any additional filters.

7. The appropriate package filters are applied and the information/product is verified that it can be sent across the security boundary.
8. One of the following occurs (dependent upon if the data is formatted message data or product files with metadata):
    a. Formatted Message Data: After the filters have been applied, the XML information object will be re-packaged and forwarded on to the low-side TIGER Publisher.
    b. Product Files with Metadata: The product package will then be decomposed into its component product and metadata file, then secure-copied with its metadata into the low-side IPL ingest directory.
9. One of the following occurs (dependent upon if the data is formatted message data or product files with metadata):
    a. Formatted Message Data: The low-side TIGER Publisher will publish the information with the IOTA PSQ.
    b. Product Files with Metadata: The low-side publisher will identify the new products placed into the low-side IPL and publish this information.
10. The low-side IOTA PSQ matches the information published by the low-side TIGER with the original low-side subscription and provides the information to the low-side application via a callback.
11. The Application receives the published data notification.

Figure 2-4 depicts the enumerated steps comprising the low-to-high Subscription Cancellation Data Flow. This data flow allows a subscription originally generated by a low-side user/application (via the low-to-high Subscription Data Flow) to be cancelled by a subsequent subscription cancellation request from the initiating low-side user/application.

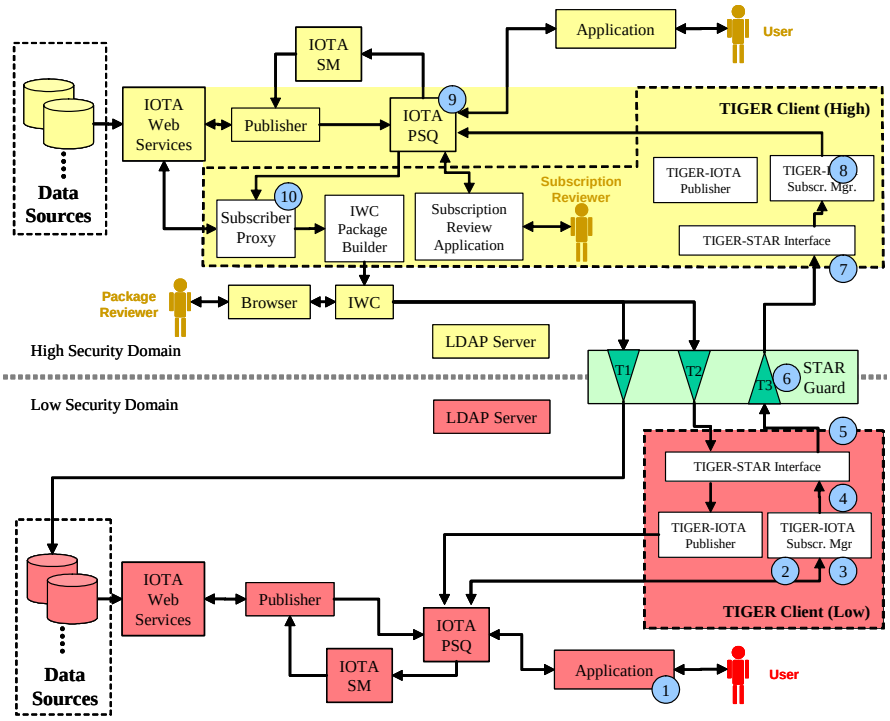

Figure 2-4: IOTA-TIGER Low-to-High Subscription Cancellation Data Flow

The following describes, in detail, the process occurring within each enumerated step shown in Figure 2-4:

12. An application in the low-side domain cancels a subscription. This involves opening a connection with the IOTA pub-sub-query (PSQ) component, publishing an IOTA subscription cancellation object (SCO) and canceling the subscription. These steps are performed within the IOTA PSQ API. Each subscription cancelled by a low-side application will also need to be marked as a cancellation for a high-side subscription and will need the credentials of the low-side users that are issuing the request added to the metadata of the information object.
13. The TIGER Application on the low-side of the STAR Guard will consist of a Subscription Manager. The Subscription Manager will register a callback with the IOTA PSQ for all IOTA SCOs.
14. The TIGER Subscription Manager will receive all SCOs as they are issued from the PSQ and will evaluate the metadata in each to verify which ones are cancellations that need to be applied to the IOTA high-side. Only those canceling subscriptions that were previously passed to the high-side should be allowed to continue on through the process.
15. All verified SCOswill be passed on to the TIGER Proxy (L) component. Each SCO will consist of an XML message, which will contain the object's metadata with a payload.
16. The TIGER Proxy (L) will then pass the SCO on to the appropriate STAR Guard thread for evaluation and dissemination across the security boundary.
17. The following processing then occurs in the STAR Guard: Each SCO received by a STAR Guard thread will be scanned for viruses prior to any other activity. Failure will terminate the transfer of the SCO across the security boundary and the SCO will be

placed in quarantine. If the SCO passes the virus scan, it will then be parsed and verified against a previously loaded XML Schema. If the XML message fails to verify according to the schema, the SCO will be dropped and transfer will be terminated. The credentials of the user and/or application initiating the SCO will be verified for authenticity. Failure in this verification process will terminate the transfer of the SCO across the security boundary. The STAR Guard thread will apply all appropriate transformations to the SCO. After all filters have been applied, the low-side signature will be removed, the information will be upgraded and resigned using the certificate of the high-side of the STAR Guard, and then forwarded on to the high-side TIGER Proxy

18. The high-side TIGER Proxy will forward the information to the TIGER high-side Subscription Manager.
19. The high-side TIGER Subscription Manager receives the SCO from the Proxy and publishes the cancellation to the high-side IOTA PSQ.
20. The high-side PSQ publishes the SCO.
21. The Client Subscription Proxy, originally created to receive information relative to the approved subscriptions, will receive notification from the PSQ that its subscription has been cancelled. Once such a notification is received, the process allocated to that subscription will be closed.

2.4.2 IOTA-TIGER Design

The following sections describe the inputs, outputs, processing, implementation approach and implementation challenges relative to each new/modified component that was developed for the IOTA-TIGER software architecture.

2.4.2.1 ISSE Web Client (IWC)

The primary purpose of the IWC within the IOTA-TIGER architecture is to provide a mechanism to ensure that a secure and completely reliable human review of IOTA information products is performed prior to releasing the information to the Guard. Due to the need of enforcing DCID 6/3 cross-domain security requirements in Top Secret/Sensitive Compartmented Information (TS/SCI) to Secret configurations, authorized review and release authorities will use the browser-based IWC user interface to review all transfers of complex information packages destined for the low-side IOTA information domain. IWC will be executed by an authorized reviewer using a COTS web browser, such as Netscape®, Mozilla® and Microsoft® Internet Explorer. IOTA information packages will be automatically queued to IWC by IOTA publish and subscribe mechanisms (i.e., the subscription Client working in concert with the IWC Package Builder) for eventual review and release to the ISSE Guard boundary control application (i.e., the STAR Guard). It should be noted that low-to-high transactions of IOTA information products can be accomplished without the need for IWC and a reliable human review.

The IWC is a Java/HTML-based software application that uses combination of Tomcat, Apache and application-developed components. It is hosted on a dedicated Sun Microsystems™ platform running Solaris™ 9 locked down for least privilege operation per National Security Agency (NSA) and Defense Intelligence Agency (DIA) guidance. From an IOTA-TIGER

implementation perspective, a few key software enhancements to the IWC were implemented to yield an IWC prototype capable of supporting the IOTA-TIGER architecture.

- IWC was modified and/or configured to support ingestion, queuing, review, and release of automatically generated IOTA information packages containing IOTA-published data in XML format and in some cases imagery data files and their associated imagery metadata.
- IWC was modified to interface to the STAR Guard. The ISSE Baseline version of IWC does not currently interface with the STAR Guard, but a prototype version developed in support of the Information Collaboration Environment for Multi-Domain AOC Networks (ICEMAN) does. This ICEMAN version was used as a starting point for the IOTA-TIGER IWC prototype capability.

2.4.2.2 Secure Trusted Automated Routing (STAR) Guard

The STAR Guard is a software system designed to securely transfer formatted data types (e.g., USMTF, XML, etc) at a high rate of speed across a security boundary. Since the data is formatted, it can be automatically filtered/sanitized and disseminated via the STAR Guard through the use of specific processing threads.

The primary purpose of the STAR Guard TIGER-IOTA threads is to perform content inspection and format validation of IOTA Information Objects and Subscription Objects prior to passing them across the boundary. There are three threads that were developed for the IOTA-TIGER architecture, each handling a different data type:

- Subscription Objects Thread: This thread reads and writes subscription objects (to include subscription cancellation messages) to and from the TIGER Client components.
- IWC Package Thread: This thread provides filters to handle IOTA-formatted message data. This thread reads its input from the IWC and writes IOTA Information Objects to the TIGER Client components.
- IWC Package & Files Thread: This thread provides filters to handle IOTA formatted message data and associated metadata and product files. This thread reads its input from the IWC and writes IOTA product files to designated directories (i.e., the IPL ingest directory) using secure copy.

All three threads leveraged prototype XML and file processing threads that were developed under the ICEMAN and Trusted Transfer Agent (TTA) efforts. Slight modification of the threads was needed to allow them to handle IOTA XML messages. For high-to -ow transfer of published information, a design decision was made regarding how best to bundle the IOTA Information Objects into the IWC Package format, (i.e., determining whether they will be contained in the message body or attached as a file to the IWC Package). Each thread validates the XML message against the appropriate, preloaded XML Schema to ensure proper format.

2.4.2.3 TIGER Client

The primary purpose of the TIGER Client within the IOTA-TIGER architecture is to facilitate the flow of information from the IOTA Publish/Subscribe System to and from the appropriate STAR Guard threads. The TIGER Client is a new component that sits between the IOTA

capabilities and a STAR Guard. It is implemented using each respective system's (i.e., IOTA and STAR Guard) currently existing API, using Java to conform to the IOTA specifications.

The TIGER Client was developed so that the same software components can be used on either the high-side or low-side of the STAR Guard. The option for high or low operation is set at runtime via one or more configuration parameters that is specified in an XML configuration file. The side of the STAR Guard that the TIGER Client software operates on, dictates the specific actions of the TIGER Client.

The TIGER Client consists of six components, the last three of which are only active on the high-side TIGER Client.

- TIGER-STAR Interface: The TIGER-STAR Interface consists of a STAR Guard client application implemented using a JNI wrapper over the STAR Guard C API. For increased security, this interface Guard occurs via mutually authenticated SSL connection. The JNI wrapped SSL-enabled STAR Guard API developed as a prototype under the ICEMAN project provides the API used as the basis for constructing this interface.
- TIGER-IOTA Publisher: The role of the TIGER-IOTA Publisher is to move data along from the TIGER-STAR Interface to the IOTA PSQ. This is accomplished via the publishing mechanism made available by the IOTA Pub/Sub Framework. It also identifies the new products placed into the receiving side IPL.
- TIGER-IOTA Subscription Manager: The TIGER-IOTA Subscription Manager has two purposes, depending on which side of the Guard it is on. On the sending side, it subscribes to subscription information objects from the IOTA PSQ component. By evaluating the metadata in each subscription, it makes a determination if it is a request for high-side information/products. Only those requesting high-side information/products are allowed to continue on to the TIGER-STAR Interface. When on the receiving side of the Guard, it reads subscription information objects from the TIGER-STAR Interface and publishes them to the IOTA PSQ component.
- Subscription Review Application (SRA): The primary purpose of the SRA is to allow a human reviewer on the high-side to examine, with the intent of approving or rejecting, each subscription as it is generated from the low-side. It accomplishes this by subscribing to low-side subscriptions from the high-side IOTA PSQ and queuing them up for the reviewer. Requests for subscriptions received by the SRA are queued up and presented to an authorized reviewer via a simple web interface. Once the human reviewer approves a particular subscription, the SRA places the subscription with the IOTA PSQ. If a request for subscription is rejected, the subscription request is deleted from queue. The SRA is a new, interactive, web-based component that is built using Java Servlets and Apache server components. It is separate from the IWC, but when operational, it will likely run on the same platform since it shares some key software components and configuration files.
- Subscription Proxy: The Subscription Proxy process is a new Java-based application component that provides an interface between the IOTA PSQ and the IWC Package Builder application. Its primary purpose is to manage all reviewed and validated low-side subscriptions requests for high-side information. The Subscription Proxy application is resident on the high-side and receives its subscription information from the IOTA PSQ by subscribing to this process for all low-side subscription requests. When information is

populated into the data sources and published to IOTA that satisfy the low-side subscription requests, the Subscription Proxy application proxys the request on behalf of a low-side user/application to receive this information and forwards it to the IWC Package Builder process.

- IWC Package Builder: The primary purpose of the IWC Package Builder is to facilitate an automated process for packaging high-side information into a valid IWC package object and posting that object to the IWC for review and release across a security boundary. The IWC Package Builder process is a new component to the IWC Application Suite that was built using components from the IUA Package Generator (IPG) and ISSE Email Gateway (IEG) (a very early prototype). The IWC Package Builder process is a Java-based application that receives its information from the Subscription Proxy application. Data received from the Subscription Proxy application consists of either an XML-based Information Object or an Imagery Product and its associated metadata. In either case, all information received by the IWC Package Builder process is in direct response to a previously approved and validated low-side subscription request for high-side information.

2.4.3 IOTA-TIGER Summary

2.4.4 Current TIGER-STAR Prototype Capabilities

Each component of TIGER-STAR system has been prototyped and is functional within a certain demonstrable scope. These components include the TIGER-STAR threads, TIGER client, Subscriber Proxy, Subscription Review Application (SRA), and IWC Package Generator (IPG.)

The components of the TIGER-STAR system have been developed to support the following data flows: low-to-high Subscription Data, high-to-low Published Information Data, and low-to-high Subscription Cancellation Data.

The low-to-high Subscription data flow is functional to the extent that users on the low-side can establish subscriptions for high-side IOTA data. These subscriptions are then processed through the TIGER-STAR system and posted in the high-side IOTA PSQ. The data flow is initiated through the following sequence of events: Subscription requests are established by a low-side user for high-side IOTA data via the low-side IOTA PSQ mechanism. The subscription requests are then passed to the low-side Subscription Manager component of the TIGER Client after it has registered with the low-side IOTA PSQ for all IOTA subscription information objects. Whenever the low-side Subscription Manager receives a subscription from IOTA, it determines which are requests for high-side information and then passes these information objects on to the TIGER Client Interface component. Once the Interface component receives the information object from the Subscription Manager, it then passes the information object on to the appropriate STAR Guard thread for evaluation and dissemination across the security boundary. When it has been determined that the subscription contains a valid request, the STAR Guard thread forwards this request on to the high-side TIGER Client Interface. The high-side Interface passes the information object on to the high-side TIGER Client Subscription Manager. The Subscription Manager receives the information object and sends the subscription information object to the high-side IOTA. Each time a subscription arrives at the high-side and has been established in the IOTA PSQ, the Subscription Proxy application registers a callback with the IOTA PSQ to receive information relative to the approved and staged subscription.

The flow to support the high-to-low Published Information data is functional to the extent that information is published on the high-side that satisfies low-side subscriptions and is transferred to the low-side based upon an acceptable reliable human review of the contents of the published data. To initiate this transfer, the high-side Subscription Proxy receives notification from the IOTA PSQ that new information is published from the high-side sources that are relative to the registered low-side subscriptions. Once the Subscription Proxy receives these notifications, it deposits these information objects into a predefined data directory to be staged for dissemination. If a notification references a product, the Subscription Proxy will order that product, along with any metadata, and stage these items for cross domain transfer by placing these files into the data directory. After the information has been staged, a user utilizing the IWC interface will then build the appropriate IWC package for review and release to the STAR Guard. Once the package is reviewed and released, the high-side information/product package is sent to the appropriate STAR Guard threads. If the package contains an information data object, it is then sent to the appropriate STAR Guard thread for automated evaluation and dissemination across the security boundary. Once it has been determined that the information object can be downgraded, the thread forwards it on to the low-side TIGER Client Publisher component. The Publisher receives the information object from the STAR Guard thread and publishes the information object to the low-side IOTA PSQ. If the package contains a product file and metadata, it is passed to the appropriate STAR Guard thread where the product file and metadata is extracted from the IWC package, evaluated, and if releasable, ftp'd into a low-side repository's source ingest directory, such as IPL.

The low-to-high Subscription Cancellation data flow is functional in that users on the low-side can cancel subscriptions for high-side IOTA data, and these cancellation notifications are then processed through the TIGER-STAR system and posted in the high-side IOTA PSQ. To support this data flow, the TIGER Client Subscription Manager component registers with the IOTA PSQ for all published subscription cancellation objects. When the Subscription Manager receives a cancellation object, it evaluates the metadata to determine if the cancellations are for subscriptions that were subscribing to high-side data. The Subscription Manager component then passes the cancellation object to the TIGER Client Interface component. The Interface component forwards all cancellation objects on to the appropriate STAR Guard thread. The cancellation object is then evaluated and disseminated across the security boundaries. The high-side TIGER Client Interface receives the cancellation object from the STAR Guard thread and passes the object to the high-side TIGER Client Subscription Manager component. The Subscription Manager receives the cancellation object and publishes the cancellation to the high-side IOTA PSQ. The Subscription Proxy, which was originally registered to receive information for the original subscription, now receives notification that the subscription has been cancelled and closes the process allocated to the original subscription.

2.4.5 Current TIGER-STAR Prototype Limitations

During the TIGER-STAR system development and the integration of the system in the lab environment at AFRL, several issues emerged that negatively impacted the development and integration efforts. Since the issues are unresolved to date, they represent limitations of the current TIGER-STAR prototype.

During integration of the TIGER-STAR system in the laboratory environment at AFRL, the TIGER Client experienced problems connecting to the Weblogic application server. These were

related to the user login/authentication module of the IOTA/JBI system. During the development phase of the effort, the JBoss server, recommended by the IOTA group, was used. The problems encountered during integration with Weblogic were not experienced with Jboss. Several attempts were made to resolve the Weblogic-related problem, but were unsuccessful. So that the integration could proceed, a JBoss application server was setup so that the TIGER Client could establish a connection to this application server.

A Subscription problem was encountered during the development effort of the TIGER-STAR system, which remains unresolved. Specifically, any subscriptions received from the IOTA PSQ containing specific geographic coordinate constraints were unable to be established. When no geographic coordinates were specified, the system was delivering the appropriate IOTA output objects. Several efforts were made to research the reasons why the geographic coordinates in subscription objects could not be established, but were unsuccessful. To avoid this problem, only those subscriptions without geographic coordinates should be used to demonstrate and test the IOTA publish and subscribe process.

Due to time constraints, several components of the original design are not part of the demonstrable architecture. These include the IWC Package Generator and a valid high-side and low-side source. A prototype component of the IWC Package Builder was developed and is operational in the development environment. Due to time constraints, the IWC Package Builder was not installed as part of this system in the lab environment. Also due to time constraints, the TIGER-STAR prototype was only able to be tested and demonstrated using Message Source Simulators as the high-side and low-side sources of intelligence data. Since it is unknown how accurately the source simulators simulate real sources (e.g., IPL), additional issues may result when these simulators are replaced with real systems.

3. Isaiah

Isaiah was developed to meet time sensitive needs at the 480th IW. Isaiah leveraged existing Air Force capabilities to provide rapid and low risk development and fielding and provided intelligence support during OEF and OIF. The specific Isaiah requirements being addressed include the capability to:

- Provide automated mechanisms for making information form data sources available on User Web sites at both SCI and collateral classification levels
- Rapidly post new information from user data sources on to Web sites
- Maintain the existing essential look and feel, and functionality of the current Web sites, including the user interface that supports queries by BE number, geographic area, date, and country code
- Provide links to Web sites with additional information
- Enhance the integrity and responsiveness of the SCI and collateral Web servers
- Provide for automatic purge of data on the Web sites with a user configurable time limit
- System must handle current data source information product data types, such as Thumbnails, JPEG, MPEG2, IPIRs, MOCSUMs, and be extensible to handle other data types

- System must interface to current SCI and collateral data sources and be extensible to handle other data sources
- System must adhere to DoD conventions and standards for representing and exchanging information, such as the DoDIIS Profile and IC Core Metadata Standard
- Handle multiple AORs
- Provide separate interfaces for overall application administration versus specific AOR administration
- Avoid reliance on BE number list coordination
- Ensure that generated web pages can be indexed and available to the ISMC search engine
- Provide separate interfaces for overall application administration versus specific AOR administration

An overview of the Isaiah architecture is presented below.

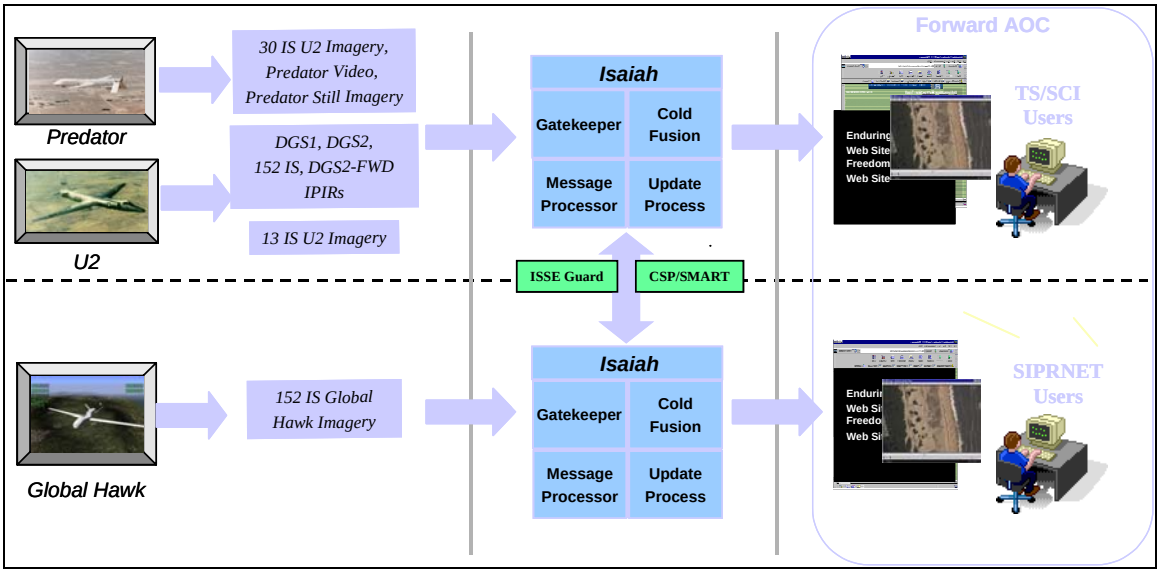

Figure 3-1: Overview of the Isaiah Architecture

Information flow through Isaiah is depicted by the graphic below.

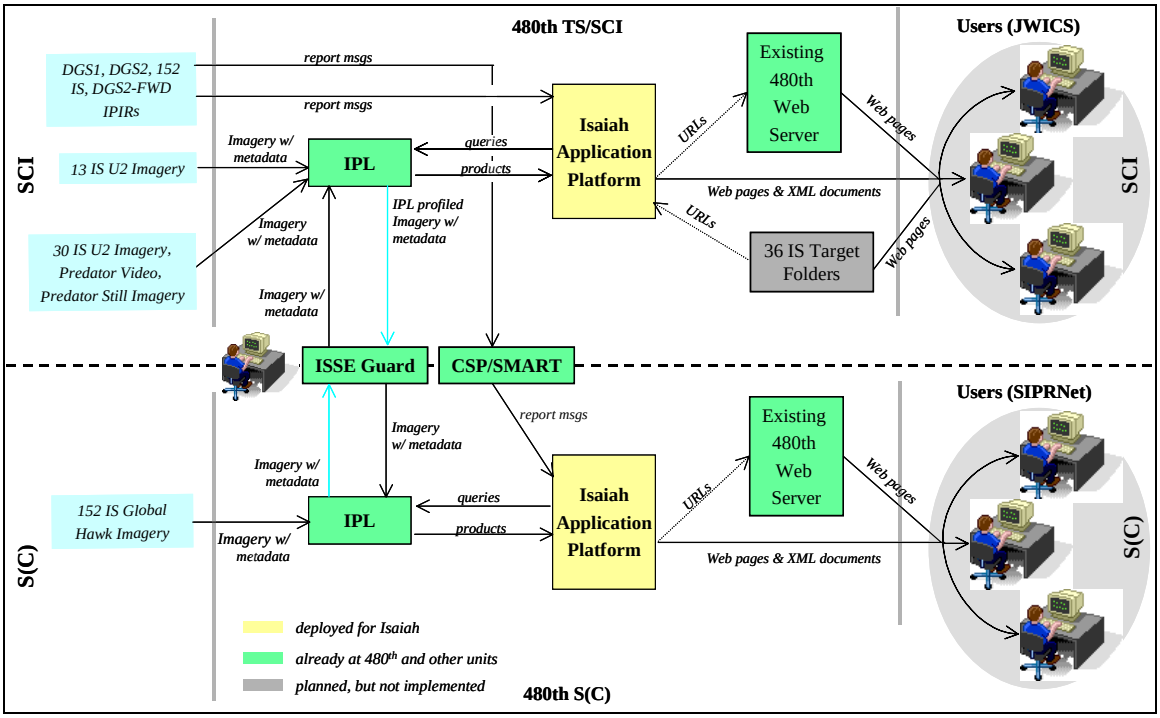

Figure 3-2: Information Flow through Isaiah

Isaiah is currently deployed at 480th IW for SIPRNet and JWICS Web sites – a sample home page is presented below.

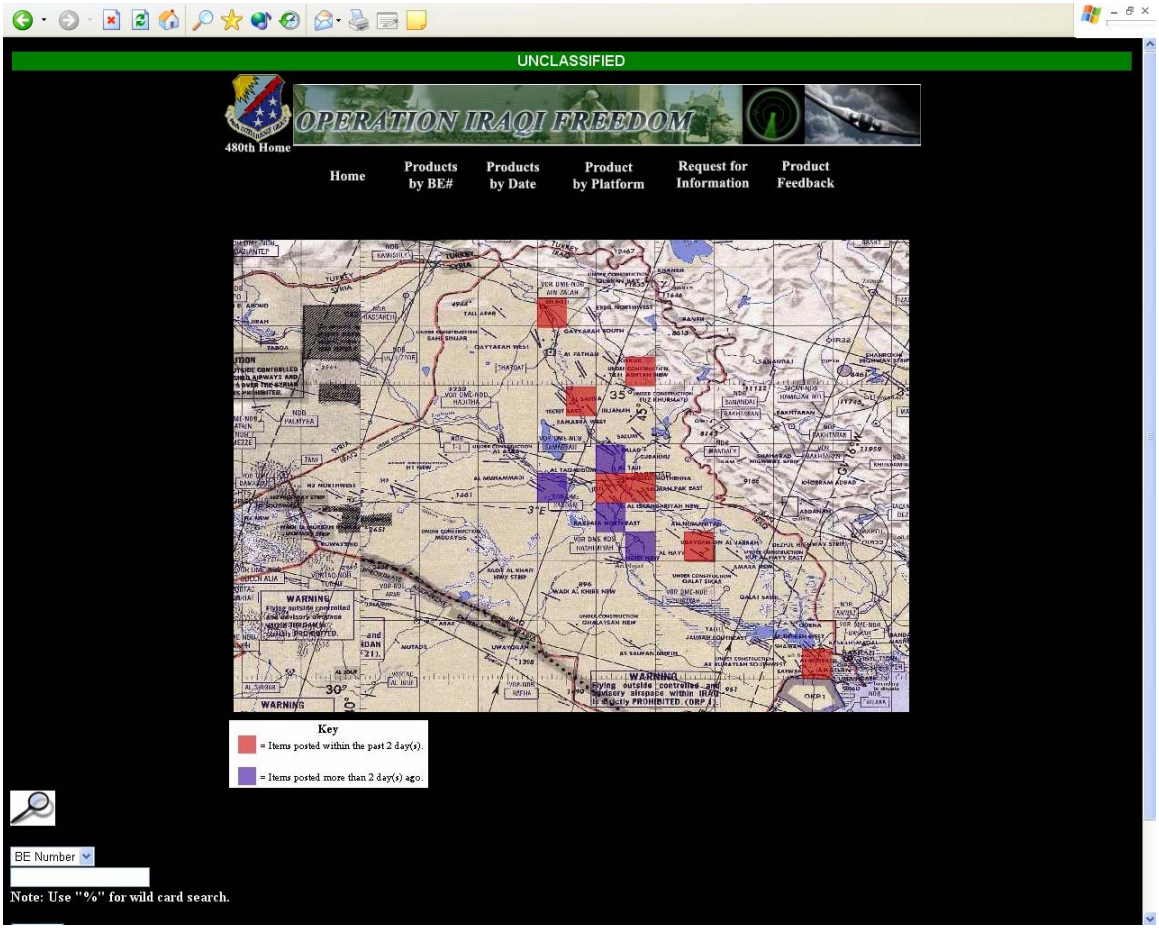

Figure 3-3: Sample Website Home Page

This deployment saved 10-15 people per DGS site supporting Web-based dissemination (see table below) resulting in a 30% staff reduction for 480th web page team. This savings allows site administrator to generate additional AORs and improved dissemination time for integrated products.

	Old Web Development	Isaiah
Imagery	Up to 20 minutes	0-5 minutes
Reports	Up to 48 hours	0-5 minutes

4. TASK 4 IOTA DOCUMENTATION

Documentation developed by the IOTA team as part of the development process has been submitted to AFRL/IFEB Configuration Management. This documentation includes:

Document	Date Submitted
System Installation Guides	20 August 2004

Document	Date Submitted
Version Description Documentation	20 August 2004
Verification Test Procedures	20 August 2004
Interface Control Documentation	20 August 2004
Security Accreditation Documentation	20 August 2004

5. REFERENCES

This section provides a listing of directives, manuals, and other documents used as reference material. References include:

1. Director of Central Intelligence Directive (DCID) 6/3, Protecting Sensitive Compartmented Information Within Information Systems, 31 Mar 01
2. DoD Intelligence Information Systems (DoDIIS) Security Certification and Accreditation Guide, April 2001, DS-2610-142-01
3. Joint DoDIIS/Cryptologic SCI Information Systems Security Standards, Revision 2, 31 March 2001
4. User Manual for the Air Force DoDIIS Infrastructure (AFDI) Unix Segment Version 1.1.0.1, Logicion-Sterling Federal, 2000
5. Common User Baseline for the Intelligence Community (CUBIC) Configuration Management Plan Version 3.0, August 2001
6. IOTA 1.0 System Security Authorization Agreement, July 2004
7. IOTA 1.0 System Installation Guide, July 2004
8. DoD Joint Technical Architecture, version 4.0, 2 April 2001
9. DoDIIS Enterprise Architecture, version 1.0, 28 May 2002
10. Global Information Grid (GIG) Capstone Requirements Document, JROCM 134-01, 30 August 2001
11. Infrastructure Operations Tools Access (IOTA) Functional Requirements Document (FRD) DRAFT 10 November, 2003
12. Geospatial and Imagery Access Services (GIAS) Specification Version 3.5, N0101-G, 06 August 2001
13. USIGS Common Object Specification Version 1.5.1a, N0104-G, 06 August 2001

6. GLOSSARY OF TERMS

Acronym	Description
5D	Demand Driven Direct Digital Dissemination
A2IPB	Automated Assistance with Intelligence Preparation of the Battlespace
AC2ISRC	Aerospace Command and Control & Intelligence, Surveillance, Reconnaissance Center
AFDI	Air Force DoDIS Infrastructure
AIS	Automated Information System
AMHS	Automated Message Handling System
AOC	Air Operations Center
AODB	Air Operations Data Base
AOR	Area of Responsibility
API	Application Program Interface
ASAS	All-Source Analysis System
BE	Basic Encyclopedia
CA	Certificate Authority
CDE	Common Desktop Environment
CD-ROM	Compact Disk Read-Only Memory
CES	Core Enterprise Services
CMDB	Configuration Management Data Base
COE	Common Operating Environment
COI	Community of Interest
COLISEUM	Community On-Line Intelligence System for End Users and Managers
COP	Common Operational Picture
COTS	Commercial Off-The-Shelf
CR	Change Request
CSP	Configuration Support Processor
CUBIC	Common User Baseline for the Intelligence Community
DAA	Designated Approving Authority
DCID	Director of Central Intelligence Directive
DCID	Director of Central Intelligence Directive
DCID	Director of Central Intelligence Directive
DIA	Defense Intelligence Agency, Bolling Air Force Base, Maryland
DII	Department of Defense Information Infrastructure
DISA	Defense Information Systems Agency
DMB	DoDIIS Management Board
DMS	Defense Message System
DoD	Department of Defense
DoDIIS	Department of Defense Intelligence Information System
EAI	Enterprise Application Integration
FIRES	Facilities Infrastructure Engineering System
FTP	File Transfer Protocol
GCCS	Global Command and Control System
GCCS I3	GCCS Integrated Imagery and Intelligence
GIG	Global Information Grid

Acronym	Description
GIG ES	GIG Enterprise Services
GOTS	Government Off-The-Shelf
GUI	Graphical User Interface
HTTP(S)	Hypertext Transfer Protocol (Secure)
HTTPD	Hypertext Transfer Protocol Daemon
IC	Intelligence Community
ICML	Intelligence Community Markup Language
IDL	Interface Definition Language
IESS	Imagery Exploitation Support System
IMO	INTELINK Management Office
IOTA	Infrastructure Operations Tools Access
IPL	Image Product Library
ISMC	INTELINK Service Management Center
ISSE	Information Support Server Environment
ISSO	Information System Security Officer
ITS	GCCS I3 Imagery Transformation Services
JAX	Java API for XML
JDCSISSS	Joint DoDIIS/Cryptologic SCI Information Systems Security Standards
JEDI	Joint Enterprise DoDIIS Infrastructure
JITF	Joint Integration Test Facility
JIVA	Joint Intelligence Virtual Architecture
JMS	Java Messaging Service
JPEG	Joint Photographic Experts Group
JTA	Joint Technical Architecture
JTT	Joint Targeting Toolbox
M3	Multimedia Message Manager
MDITDS	Migration Defense Intelligence Threat Data System
MEPED	Military Equipment Parametric and Engineering Data Base
MIDB	Modernized Integrated Database
MTIX	Moving Target Indicator
NCES	Network Centric Enterprise Services
NIMA	National Imagery and Mapping Agency
NITF	National Imagery Transmission Format
PKI	Public Key Infrastructure
PR	Problem Report
RMS	Requirements Management System
SAT	Site Acceptance Testing
SCI	Sensitive Compartment Information
SDE	Support Data Extension
SDK	Software Development Kit
SIMG	System Installation & Maintenance Guide
SOAP	Simple Object Access Protocol
SPIA	Standards Profile for Imagery Archives
SQL	Structured Query Language
SRTM	Security Requirements Traceability Matrix

Acronym	Description
SSAA	System Security Authorization Agreement
SSL	Secure Sockets Layer
TBMCS	Theater Battle Management Core System
TFM	Trusted Facility Manual
TIBS	Tactical Information Broadcast Service
TIFF 6.0	Tagged Image File Format 6.0
TRAP	TRE Related Applications Program
TRAP/TRE	Tactical Related Applications/Tactical Receive Equipment
TRE	Tactical Receive Equipment
TTA	Trusted Transfer Agent
UDDI	Universal Data Discovery Interface
URL	Uniform Resource Locator
WSDL	Web Services Definition Language
WSIF	Web Services Invocation Framework
WX	Air Force Weather
XML	eXtensible Markup Language